

## Simulating Nios II Designs with Program Memory on External Flash and SSRAM

As described in [AN 351: Simulating Nios II Embedded Processor Designs](#), Qsys and Nios II Software Build Tools (SBT) for Eclipse can automatically generate an RTL simulation environment for Nios II designs. This guide shows how to handle issues specific to designs in which program memory is stored on external flash memory and runs from external SSRAM memory. This guide is presented using Verilog, but the process is similar for designs using VHDL.

### Before You Begin

This guide assumes that you have prior experience using Qsys as well as a familiarity with Nios II SBT and the ModelSim simulator. In order to simulate the Nios II design using the instructions in this guide, you must have the following software installed:

- The Quartus® II software version 11.0 or later
- ModelSim-Altera Edition version 6.6d or higher
- Nios II Embedded Design Suite version 11.0 or later

### Design Description

The design used for this guide is a complete Qsys system that includes a Nios II processor, connections to external flash and SSRAM memory, and PIOs for buttons and LEDs. The example software running on the NIOS II system tests a range of addresses in SSRAM and signals possible write/read failures by turning on the LEDs. Additionally, messages are sent to the console via the JTAG UART to confirm success or failure of each address.

### Generating the Simulation Files

This section summarizes the process of generating the simulation files. A more detailed explanation can be found in [AN 351: Simulating Nios II Embedded Processor Designs](#).

#### Setting Up and Generating the Simulation Environment in Qsys

1. Download the **niosii\_with\_external\_memory.zip** design example listed in the [Downloads](#) section below. Extract the design example to a new directory. This directory will be referred to as **<project directory>** throughout the remainder of this guide.
2. Start the Quartus II software, and select **File > Open Project**. Browse to **<project directory>**, select the **niosii\_with\_external\_memory.qpf** file, and click **Open**.

3. Open Qsys by selecting **Tools > Qsys**. Open the **niosii\_system.qsys** file.
  - The design example used for this guide is a complete Qsys system. If you are using your own Qsys design, ensure that your design is complete before you start to generate simulation models.
4. On the Generation tab, set the following parameters to these values.
  - **Create simulation model—None**
  - **Create testbench Qsys system—Standard, BFM for standard Avalon interfaces.**
  - **Create testbench simulation model—Verilog**
  - **Create HDL design files for synthesis—Turn off**
  - **Create block symbol file (.bsf)—Turn off**
5. Click **Generate**. Save the system if prompted.

### Creating the Nios II Software

The software used for this guide will test a range of address in SSRAM and signal success and failure with the LEDs and the JTAG UART. If a read/write failure occurs, the LEDs will turn on. Otherwise, the LEDs will remain off. With each test, messages are sent to the console via JTAG UART to confirm success or failure. To create and build the software project, perform the following steps:

1. Open Nios II SBT for Eclipse version 11.0 or later, and select **File > New > Nios II Application and BSP from Template**.
2. Select the SOPC Information File by browsing to *<project directory>* and selecting **niosii\_system.sopcinfo**.
3. For Project Name, type **"niosii\_with\_external\_memory"**.
4. Select **Blank Project** from the Templates options.
5. Click **Finish**.
6. Next, import the **niosii\_with\_external\_memory.c** source file into the software project by using the Eclipse Import dialog or by copying the source file directly to the software directory:  
*<project directory>/software/niosii\_with\_external\_memory/*

### Optimizing the Simulation for External Memory

When simulating Nios II systems that use external memory, it is often desirable to skip the simulation of the transfer of program code from non-volatile storage to RAM. Nios II SBT provides a simulation optimization option that skips this step and preloads the RAM with program code. To enable this optimization:

1. Right-click on **niosii\_with\_external\_memory\_bsp** in the Project Explorer and select **Nios II > BSP Editor....** As shown in Figure 1, the BSP Editor will open with the Main tab selected and **Settings > Common** selected from the hierarchy on the left side of the window.

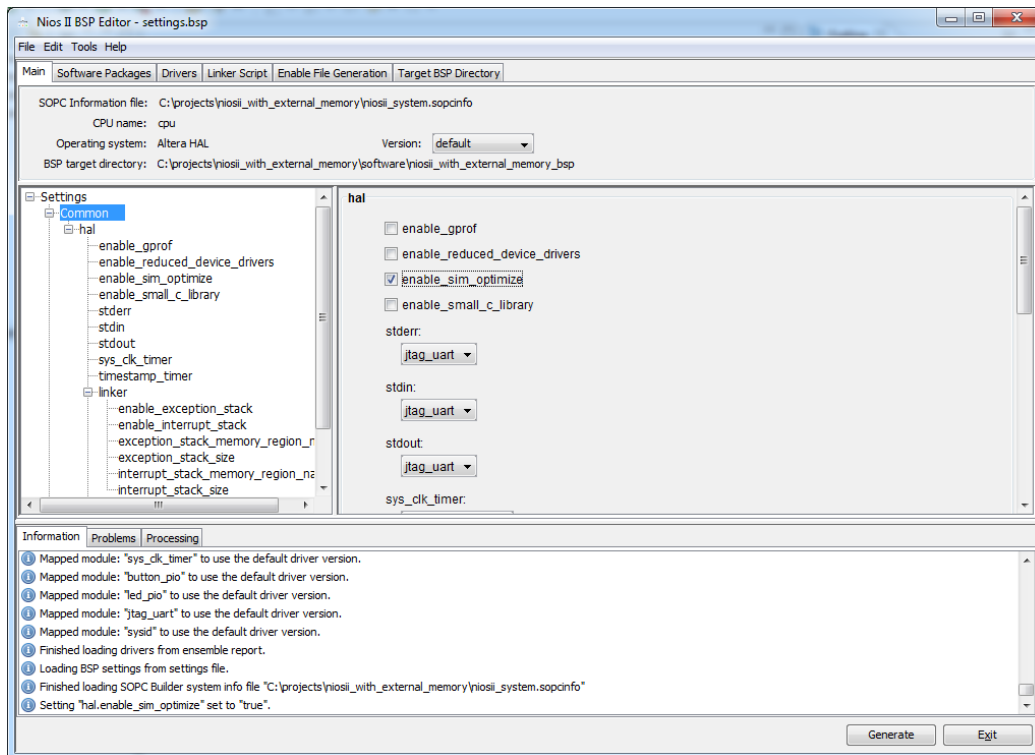


Figure 1 – Selecting "enable\_sim\_optimize" to optimize the simulation for external memory.

2. As shown in Figure 1, check the box next to the **enable\_sim\_optimize** option.
3. Click **Generate**.
4. When the BSP generation has completed, click **Exit** to return to the main Nios II SBT window.

**Important note:** when the **enable\_sim\_optimize** option is turned on, the BSP cannot run on hardware. To run your project on hardware, disable the **enable\_sim\_optimize** option by unchecking the box in the BSP Editor.

## Building the Project

- If you used the Eclipse Import dialog to include the source file, build the project by right-click on **lower\_level\_qsys** in Project Explorer and select **Build Project**.
- If you copied the source file directly into the software directory, do a clean build by selecting **Clean Project**.

## Fixing the SSRAM and Flash Bus Functional Models

Due to a known issue with the SSRAM and flash memory bus functional models (BFMs), a slight change must be made to the automatically generated testbench file before you run your simulation. For this guide, the testbench file, **niosii\_system\_tb.v**, is located in:

**<project directory>/niosii\_system/testbench/niosii\_system\_tb/simulation**

In general, this file will be located in:

**<project directory>/<Qsys system name>/testbench/<Qsys system name>\_tb/simulation**

For this example design, you can either make the modifications to the test bench file yourself, or you may copy the completed code into the testbench file from the **niosii\_system\_tb.v** file, which is located in **<project directory>/modified\_files** directory.

To make the modifications, open the automatically generated testbench file in an editor of your choice and complete the following steps:

1. As shown below, add a new wire called **system\_clk** and connect it to the **system\_clk** port of the Nios II system:

```
// CHANGE: add system_clk wire and connect to system_clk port of Nios II system
wire system_clk;

niosii_system_tb_niosii_system_inst niosii_system_inst (
    .osc_clk          (niosii_system_inst_osc_clk_clk_in_bfm_clk_clk),
    .system_clk       (system_clk),
    ...

```

2. As shown below, connect the **system\_clk** wire to the **clk** input of the SSRAM BFM (in place of the **ext\_ssram\_external\_mem\_bfm\_clk\_bfm\_clk\_clk**).

```
// CHANGE: connect the system_clk wire to the clk input of the SSRAM BFM (in place of
ext_ssram_external_mem_bfm_clk_bfm_clk_clk(
...
    ) ext_ssram_external_mem_bfm (
        .clk (system_clk),
    ...

```

3. Next, connect the **system\_clk** wire to the **clk** input of the flash memory BFM (in place of the **ext\_flash\_external\_mem\_bfm\_clk\_bfm\_clk\_clk**).

```
// CHANGE: connect the system_clk wire to the clk input of the flash memory BFM (in
place of ext_flash_external_mem_bfm_clk_bfm_clk_clk(
...
    ) ext_flash_external_mem_bfm (
        .clk (system_clk),
    ...

```

4. Save the **niosii\_system\_tb.v** file.

## Running the Simulation

Now that you have finished fixing the testbench file, go back the Nios II SBT and right-click again on “**lower\_level\_qsys**” in Project Explorer, and select **Run As > Nios II ModelSim**. This command will start the ModelSim software and create the remaining simulation files.

Once ModelSim has finished loading the simulation files, load signals into the waveform viewer by selecting **File > Load...** and browsing to the **<project directory>/wave.do** waveform script file. Begin the simulation by typing “**run 2 ms**”. As shown in Figure 2, the software tests several SSRAM address and displays success messages for each test. The active-low LEDs remain off (0x1111) for the duration of the test to indicate that all of the tests have been successful.

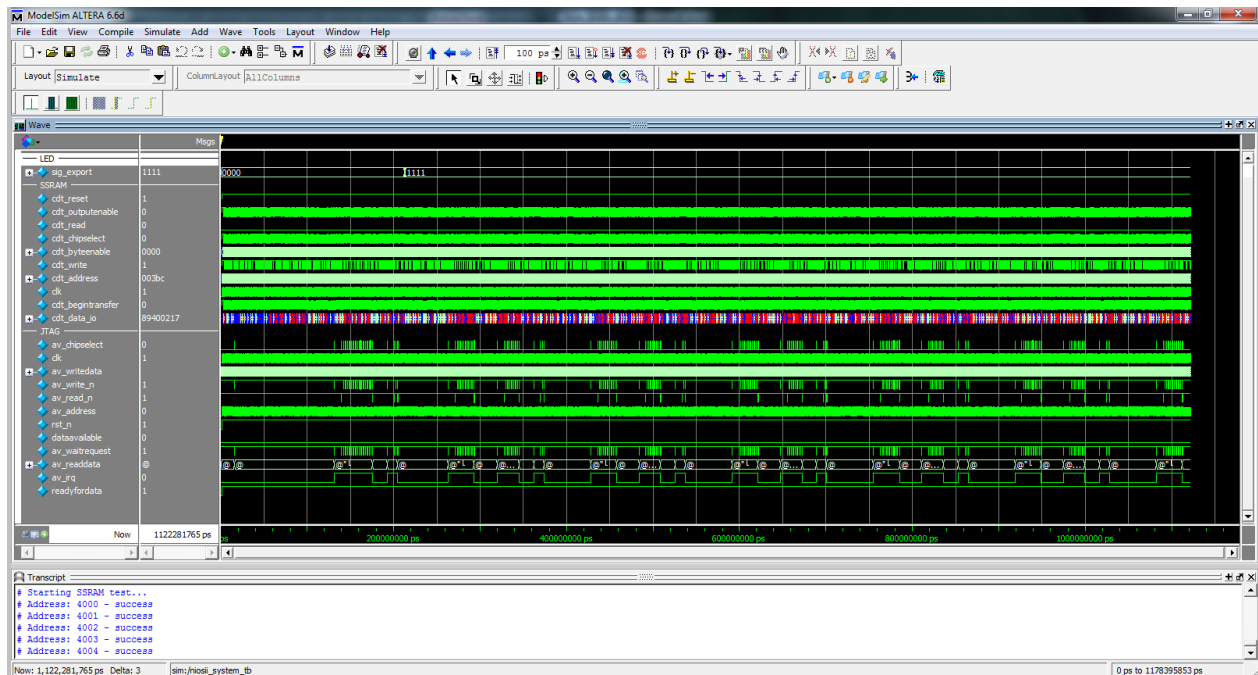


Figure 2 – Simulating the niosii\_with\_external\_memory design.

## Additional Resources

- [AN 351: Simulating Nios II Embedded Processor Designs](#)
- [Nios II Software Build Tools Reference](#) – Nios II Software Developer’s Handbook, Sec. IV, Ch. 15.